1. Premessa

In molte aree della matematica computazionale geometrica/numerica applicata, tra cui la geo-mappatura, la visione artificiale,la grafica computazionale, l’ analisi degli elementi finiti, l’imaging medico, il disegno geometrico e la modellazione solida, bisogna calcolare le incidenze, le adiacenze e l'ordinamento delle celle, generalmente usando disparati e spesso incompatibili strutture dati e algoritmi.

Questo documento introduce algoritmi di topologia computazionale per scoprire le partizioni spaziali 2D/3D indotte da un insieme di oggetti geometrici di dimensione rispettivamente 1D/2D. I metodi e il linguaggio sono quelli della topologia geometrica e algebrica di base. Vengono utilizzati solo vettori e matrici sparsi per calcolare sia gli spazi che le mappe, cioè il complesso della catena, dalla dimensione zero alla dimensione tre.

Il software prototipo è scritto in Julia, il nuovo linguaggio per l'informatica scientifica. Le applicazioni possono variare dalla grafica 3D alla stampa 3D, dalle immagini alla comprensione della scena, dai giochi alla modellazione delle informazioni edilizie (BIM).

1.INTRODUZIONE

Data una collezione S di oggetti geometrici, l'oggetto di questo articolo è il calcolo della topologia della loro disposizione spaziale A(S) come un complesso di catene, cioè come una breve sequenza esatta di spazi lineari Ci di co-catene e mappe lineari di confine/co-confine ∂p e δp = ∂ p+1 fra loro:

C• = (Cp, ∂p ) := C3 ↔ C2↔ C1 ↔C0 .

Il complesso della catena C• caratterizza pienamente la topologia della partizione spaziale (arrangiamento) indotto all'interno dello spazio euclideo ambientale da una collezione di oggetti geometrici incorporati in esso. Per costruire C•, la raccolta di dati oggetti geometrici (complessi di celle) deve essere unita in una struttura comune. Le strutture dati necessarie per tale programma computazionale sono multi-array sparsi e le loro operazioni algebriche standard. In questo modo, introduciamo un nuovo approccio basato su una topologia algebrica lineare a tratti, che permette di trattare una topologia cellulare piuttosto generale di complessi, con celle omeomorfe ai poliedri, cioè agli spazi triangolari, e quindi possibilmente non convesso e multi-connesso. Crediamo che la topologia algebrica geometrica di base fornisca già il giusto insieme di concetti e strumenti per calcolare ed esplorare le celle della partizione spaziale indotte da un insieme di oggetti geometrici, nonché le relative relazioni incidenza/vicinato. L'attuale escalation di quantità/qualità dei dati, e la loro deriva attraverso “tubature” di micro/macro-servizi che necessitano interfacce semplici, insieme alla rapida diffusione di architetture ibride per applicazioni avanzate, è anche spiegato in questo documento. Le nozioni di cui ci occupiamo includono complessi geometrici, spazi lineari di catene e co-catene, il complesso a catena degli operatori lineari tra coppie di spazi, e le loro composizioni.

La discussione è limitata alla topologia lineare a tratti e alle dimensioni dello spazio minore o uguale a tre. Il documento formalizza gli algoritmi per generare le matrici di operatori co-confine su una disposizione cellulare nello spazio euclideo. Per i complessi simpliciali, cioè le triangolazioni, gli operatori al contorno sono definiti come estensioni lineari degli operatori di confine di base che agiscono sui simplessi. Il modo standard per esaminare le strutture dati combinatorie è la struttura dati IG (Grafico di incidenza) ed è un'implementazione del diagramma di Hasse delle celle di un complesso d. Il nostro complesso di co-catena fornisce un rappresentazione algebrica del diagramma di Hasse con matrici sparse, associando ciascuna a due livelli adiacenti con mappe di co-confine per attraversare su e giù la gerarchia. Sono stati sviluppati molti altri sistemi circa tre decenni fa, per gestire la fusione di complessi nel contesto della modellazione solida e automazione della produzione. Il Complesso Geometrico Selettivo (SGC) è stato introdotto nel 1989, considerando celle di dimensioni inferiori contenute all'interno di celle principali. Successivamente, gli scienziati nel Regno Unito hanno proposto l'API Djinn , con le basi teoriche per operazione di fusione di complessi cellulari, ma le rappresentazioni di confine (non molteplice) sono rimaste lo standard nel campo. Recentemente, è stata proposta una procedura sistematica per la costruzione di una famiglia di maglie costruttive esatte. La maggior parte degli algoritmi e delle procedure precedenti lavorano con strutture dati ottimizzate per classi selezionate di oggetti geometrici. Al contrario, la nostra formulazione, rappresentazione e algoritmi, espressi in termini di complessi di co-catene di mappe co-confine, possono essere applicati a oggetti geometrici molto diversi, che vanno dai modelli solidi alle maglie ingegneristiche, ai sistemi geografici, alle immagini biomediche. Metodi numerici che mirano a integrare modelli di dominio, topologia differenziale e matematica anche la modellizzazione con simulazioni fisiche si basa su catene e co-catene. In particolare, il metodo cellulare (CM) è un metodo di calcolo puramente algebrico per la modellazione e la simulazione basato su mappe di confine/co-confine e formulazione diretta discreta di leggi di campo. Tutti gli algoritmi di unione, per natura, seguono gli stessi passaggi. I vantaggi di formularli in termini dei complessi di co-catene e delle operazioni su matrici sparse sono i seguenti:  
(1) si rivela la natura topologica algebrica (comune e generale) di questa operazione;

(2) i dettagli e gli algoritmi di basso livello sono nascosti;

(3) è fornita una connessione esplicita ai kernel SpMV (moltiplicazione tra vettori e matrici sparse) e ai sistemi di algebra lineare numerica sparsa sulle piattaforme GPU e HPC;

(4)è supportato lo sviluppo sistematico e rigoroso degli algoritmi, corretti per costruzione.

**2. Pipeline computazionale**

Iniziamo con una collezione di input S di complessi di celle lineari a tratti di (d − 1) dimensione, incorporati in uno spazio Ed, con d ∈ {2, 3}: sono possibili insiemi di linee o poligoni, superfici triangolari , composte da maglie cubiche, 1, 2 o 3 celle da elementi di immagine 2D o 3D (pixel o voxel, rispettivamente), 2-scheletri/confini di poliedri triangolati, B-reps non molteplici o ripetizioni decompositive di modelli solidi. Questi oggetti sono complessi geometrici, cioè coppie (X, µ), dove X è un complesso cellulare che specifica la topologia e µ : X0 → Ed è la funzione di inclusione delle 0-celle, sufficiente per una geometria lineare a tratti. I dati possono contenere sia (d − 1) che d-complessi: l'unione combinatoria dei loro (d − 1)-scheletri è selezionata come input effettivo alla pipeline. Un insieme di input ammissibile S di complessi geometrici si intersecheranno e manderanno Ed in un complesso cellulare X = ∪ Xp (0 ≤ p ≤ d), detta disposizione A(S) indotta da S. L'oggetto di questo lavoro è il calcolo del complesso di catena C•(X) = (Cp, ∂p ), partendo da qualche rappresentazione di S. In particolare, calcoliamo le matrici delle mappe lineari ∂p (e le loro duali δp−1) tra gli spazi di catena Cp . Poiché la matrice di una mappa lineare Cp → Cp−1 tra spazi lineari contiene per colonne lo spazio che rappresenta degli elementi di base dello spazio del dominio, il documento fornisce anche algoritmi costruttivi per generare una rappresentazione matriciale sparsa degli elementi di base fino a Cp , che sono uno a uno con p-celle negli scheletri di Xp. Osserviamo che le celle in X, in particolare in Xd , non sono note in anticipo. Il calcolo è corretto perché i confini delle 2 celle scomposte adiacenti sono compatibili come complessi cellulari per costruzione. Un requisito della definizione standard di un complesso di celle richiede la compatibilità dei confini per essere mantenuto. Questo fatto è garantito dal fatto che sottoinsiemi adiacenti di 2 celle che hanno intersezione non vuota, generano 0 e 1 celle congruenti sul loro confine comune.

**2.1 Disposizioni 2D generate da 2 celle (Merge)**

Sia S2 ⊆ S l'insieme di 2 celle di complessi geometrici di input , incorporato in E. Osserviamo che non è richiesto che S2 sia un complesso di celle, poiché le celle possono intersecarsi al di fuori dei loro confini. È solo richiesto che ogni cella sia connessa e molteplice. Ogni σ ∈ S2 è mappato al sottospazio z = 0 da una trasformazione affine Qσ , insieme all'insieme I(σ) ⊂ S2 di celle che potenzialmente lo intersecano. L’ insieme Σ = X2(σ ∪ I(σ)) è intersecato con il sottospazio z = 0 , producendo un insieme S1(σ) di segmenti di linea in E2 . In primo luogo, questi sono intersecati tra loro, producendo il complesso di catena C•(σ) = (C2,1,0, ∂2,1) generato da A(S1). È necessario prestare attenzione per identificare gli 1-cicli intorno ai fori all'interno di 2-celle partizionate, al fine di rimuovere i loro cicli di confine esterni, rimuovendo le loro colonne e il ciclo esterno dalla matrice dell'operatore. L'identificazione è facile: ogni foro ne produce due colonne opposte che si sommano a 0. Infine, il 2-complesso geometrico Xσ viene ritrasformato in E3 da Q-1 . In sintesi, l'algoritmo 1 esegue la mappa uno a uno σ → Xσ , calcolando le mappe σ → C•(σ) indipendentemente l'uno dall'altro. L'output è un insieme C := {C•(σ), σ ∈ S2}.

**2.2 Calcoli dell'insieme dei quozienti (congruenza)**

L'idea che ci permette di calcolare frammentazioni indipendenti di 2 celle deriva da una similitudine tra omologia e congruenza. Due (d − 1)-spazi (curve, superfici, ecc.) incorporati in E D sono topologicamente omologhe quando i loro confini possono essere incollati, racchiudendo una porzione dello spazio ambiente e suddividere E D in due parti, interna ed esterna. Due figure geometriche sono geometricamente congruenti se l'una può essere trasformata nell'altra mediante un'isometria. Le congruenze Rp tra p-celle di complessi geometrici in sono relazioni di equivalenza, quindi possiamo calcolare il complesso di catene di quozienti di catene spaziali, C2(U2/R2) 2 −→ C1(U1/R1) 1 −→ C0(U0/R0), su cui successivamente costruire la base ancora sconosciuta di C3. Nota che Up = D U P , con σ ∈ S2, è l'unione di basi di “p-catene frammentate ” in ƒ, modulo le relazioni di congruenza Rp . Cp (Su/Rp ) sta per lo spazio della catena generato da Xp = Up /Rp . In questa fase della pipeline computazionale, calcoliamo, per ogni σ ∈ S2, gli insiemi quoziente e le mappe ∂p intermedie, per p = 0, 1, 2. Come di consueto, si procede con un procedimento induttivo: ogni fase consiste nell'incollare celle di dati dimensione al risultato della fase precedente . La costruzione della matrice sparsa del operatore con segno ∂1 : C1 → C0 è semplice: per ogni u h 1 = u k2 0 − u k1 0 , basta scrivere ∂1[k2,h] = 1 e ∂1[k1,h] = −1, per convenzione per k2 > k1. Per i dettagli delle operazioni sui quozienti, vedere la Sezione 3.2.

Qui discutiamo del risultato quando la collezione di input è definita da una coppia di complessi adiacenti con incompatibili sub-complessi lungo la loro interfaccia. Un semplice esempio può includere due cubi unitari tetraedrici che sono incidenti su una faccia planare triangolata in due triangoli ma lungo diagonali diverse. Analogamente al caso 2D, ogni triangolo di confine verrebbe frammentato contro tutti quelli incidenti, producendo facce di output frammentate, in modo che il risultato sul supporto affine comune (diciamo, il piano verticale) sarebbe esattamente quattro triangoli, perché ogni triangolo di input è frammentato dall'altra diagonale. Un esempio più grande rappresentato dalla disposizione 2D generata da un numero di segmenti di linea casuali.

**2.3 Confezione regalo topologica (TGW)**

L'algoritmo discusso qui viene utilizzato per calcolare topologicamente in 2D/3D, rispettivamente, le matrici sparse di operatori con segno ∂2 e ∂3, a partire rispettivamente dall'ingresso ∂1 e ∂2. La matrice [∂d ] della mappa lineare Cd → Cd−1 tra spazi lineari contiene per colonne lo spazio obiettivo rappresentazione degli elementi base dello spazio del dominio, come catene chiuse (d-1), cioè (d-1)-cicli. All'interno del pipeline computazionale discusso in questo documento, TGW viene utilizzato localmente per ogni 2 celle da scomporre e globalmente per generare le 3 celle della disposizione dello spazio ambientale E 3 . L'algoritmo pseudocodice è dato e discusso nella Sezione 3.3. Le prolunghe necessarie per la gestione di fori e i componenti non collegati sono descritti in dettaglio nella Sezione 3.4.2.

Il metodo topologico qui introdotto, che ricorda l'algoritmo “gift-wrapping” per il calcolo degli involucri convessi di insiemi discreti di punti 2D e 3D, è dettagliato e formalizzato in Sezione 3.3. L'algoritmo TGW prende una matrice sparsa [∂d−1] come input e produce in output a matrice sparsa [∂ + D ], aumentata con la cella esterna. Una funzione di immersione geometrica µ : X0 → E D è usata per calcolare l'ordinamento angolare, attorno ad alcune (d − 2)-celle, di (d − 1)-elementi di base nel co-confine, mentre avvolge un (d − 1)-ciclo. I cicli costruiti (minimi) sono impostati come colonne di [∂d ], nella costruzione della base Cd. Nota anche che, una volta fissati gli insiemi ordinati degli elementi di base, le colonne contengono la rappresentazione delle coordinate di (d − 1)-cicli, costruiti dai coefficienti di gruppo ({−1, 0, 1}, +) ≃ Z/3Z = Z3. Analogamente, i confini e i co-confini delle catene sono calcolati moltiplicando le matrici degli operatori per i vettori coordinati di tali coefficienti.

**2.4 Componenti non collegati (fori)**

La cella esterna della disposizione spaziale X = A(S) potrebbe avere un confine non connesso, comprendente più di un (d −1) ciclo , come una palla meno una palla concentrica più piccola. Analogamente, X potrebbe contenere sia componenti non connessi che eventualmente nidificati. L'algoritmo TGW in realtà calcola tutti i cicli al contorno, che devono essere adeguatamente gestiti per produrre un singolo Matrice [∂d ]: scomporre prima l'ingresso [∂d−1] in componenti connesse; quindi assemblare/rimuovere i cicli vuoti

**2.4.1 Decomposizione del 2-scheletro**. Consideriamo un grafo bipartito G = (N,A), con N ≃ Λ2 ∪ Λ0, e A ⊆ Λ2 × Λ0, associato alla matrice caratteristica sparsa che codifica la relazione di incidenza. G ha un nodo per ogni 2 celle, un nodo per ogni 0 cella e un arco per ogni coppia incidente. Pertanto, gli archi in G sono uno a uno con gli elementi diversi da zero della matrice A. Calcolando le componenti puntiformi massime di G, suddividiamo lo scheletro di X2 in h componenti connesse: X2 = {X P 2 }, tale che 1 ≤ p ≤ h. Per ogni componente X P 2 , ripeti quanto segue ;

Per prima cosa, assemblare la matrice sparsa [∂2] P, e calcolare il corrispondente [∂ + 3 ] p generato dall’algoritmo 2. Quindi, suddividere [∂ + 3 ] P nell'operatore di confine ∂ P 3 : C P 3 → C2 e la colonna matrice c p = ∂ + 3 [σ P ] ∈ C2 della cella esterna σ p X3. L'insieme S = {c P } di h 2-cicli disgiunti, è l’ inizializzazione del set di shell Xd . Quindi aggiungiamo all'insieme S i cicli vuoti (shell) già inclusi all'interno di qualche cellula non contrattile. Li chiamiamo fori di input. Scopriamo i fori di ingresso (non modificati) mediante ispezione diretta delle matrici [∂ P3 ], poiché i fori sono rappresentati lì come coppie di colonne a somma zero. Infatti ogni foro di input, se non intersecato da altri dati, restituisce un valore non modificato, e produce due colonne opposte nella matrice componente [∂ P3 ]. Ogni corrispondente coppia di colonne ha tutti gli elementi diversi da zero sulle stesse righe, ma con segni opposti (orientamenti). L'ispezione avviene tramite una sorta di algoritmo scan-line che lavora sulle righe di ogni [∂ P 3 ] matrice, che riconosce le coppie emergenti di colonne opposte e memorizza ciascuna coppia (foro candidato) fino a quando il suo contenuto alla fine differisce. L'algoritmo procede spostandosi da una riga alla successiva, fino a raggiungere la fila inferiore e restituire il set, eventualmente vuoto, di fori. Una colonna da ciascuna coppia (con segno appropriato, vedere 2.4.3) viene rimossa dalla matrice componente [∂ P 3 ], e il suo 2-ciclo opposto viene aggiunto all'insieme S di shell.

**2.4.3 Riduzione transitiva di shell poset.** Il rapporto di contenimento antisimmetrico tra i gusci (2-cicli) in S è calcolato per tutti, dal test di contenimento PointInShell tra un singolo vertice u e il ciclo c J 3 . Il caso 3D generale è descritto qui. Consideriamo il punto p io = µ(u io 0 ) e le 2 celle in [c J 2 ] = [∂3][c J 3 ], con µ il funzione di inclusione U0 → E 3 . Un semplice calcolo di contenimento, dove ogni punto p io è testato per contenimento contro ogni c J 2 ciclo, 1 ≤ i, j ≤ s = #S, verrebbe calcolato in tempo quadratico O(s 2 ). Un'efficiente procedura O(s log s) è invece qui abbracciata, utilizzando due (cioè, d −1) alberi unidimensionali di intervallo per le scatole di contenimento di c J 2 elementi, con 1 ≤ j ≤ s, per selezionare solo i termini in ogni [c J 2 ], le cui scatole intersecano un raggio (scatola degenerata) dal punto testato. Il contenimento con il test in 3D viene quindi eseguito intersecando il raggio da p i con i piani contenenti le 2-celle nell'output della query e testando il contenimento punto-poligono 2D nei loro piani (tramite mappe per il sottospazio z = 0). Ogni test planare punto-poligono viene eseguito in tempo lineare con il numero di bordi sul confine della 2-cella. Tutti gli elementi della colonna i di [S 2 ] sono così prodotti dopo una singola interrogazione su alberi di intervallo, considerando la parità di risposte positive ai test di contenimento. Quindi, il grafico orientato della riduzione transitiva R della relazione di contenimento S 2 viene estratto. Dal momento che S 2 è antisimmetrico, il grafico di R ridotto è un albero. Se l'insieme di bordi di questo albero è vuoto, ogni componente disgiunta di X3 è contenuta all'interno di un’ altra, e sia X3 che ∂3 possono essere assemblati per unione disgiunta di 3 cellule di X P 3 e colonne di [∂3] P , (1 ≤ p ≤ h), rispettivamente. Se, viceversa, quanto sopra non è vero, si riduce il grafico R a una foresta di alberelli annullando gli archi a distanza pari dalla radice e, per ogni arco (i, j), scopri quale cella del contenitore componente X io contiene effettivamente il componente contenuto X J , cioè il suo guscio c J 2 ed è forse internamente non vuoto. Situazioni di intersezione più complesse sono impossibili per costruzione, poiché i componenti sono a priori disgiunti. Pertanto, in caso di contenimento, un componente è necessariamente contenuto in qualche cella vuota dell'altro.

**2.4.4 Sulla robustezza dei calcoli.** I problemi di robustezza compaiono ovunque in geometria computazionale, a causa di problemi sia numerici che topologici. Nel nostro caso possiamo affermare che l'algoritmo TGW termina correttamente ogni volta che il suo input è topologicamente corretto. Questo è sempre vero in 2D per costruzione, tramite l'eliminazione dei sottografi che non sono 2-connessi. In 3D, la matrice di input [∂2] può essere o meno topologicamente corretta. È imperfetta quando contiene qualcosa di incidente a una sola 2-cella, cioè alcune colonne con un singolo elemento diverso da zero. In questo caso il TGW va in loop e non termina. Quando l'algoritmo termina, l'output [∂ + 3 ], cioè la base di 3 celle indipendenti e i cicli esterni, sono calcolati correttamente per costruzione. Con infiniti calcoli di precisione l'algoritmo TGW terminerebbe sempre correttamente. I possibili difetti di 2-scheletro (quindi di [∂2]) dipendono da errori di congruenza, cioè da confini comuni incompatibili tra 2 celle decomposte. Tali errori topologici sono generati da errori numerici. Fonti di errori numerici nella nostra pipeline può risiedere solo, per progettazione, nell'intersezione a coppie di 2D segmenti di linea, e nel limite ϵ ha permesso di identificare vertici molto chiusi. La compatibilità di confine può essere costretta con i seguenti passaggi: (a) accettazione dell'intersezione di parametri leggermente al di fuori del loro dominio [0, 1];

(b) applicazione del processo di suddivisione tramite Float128 variabili successivamente convertite a Float64;

(c) identificazione immediata dell'istanza a due punti generata;

(d) utilizzo locale di un diametro maggiore per l'identificazione numerica di vertici (quasi)congruenti.

**2.5 Efficienza della rappresentazione in array della topologia**

Notiamo che:

• le p-celle (come catene 0 o cicli p) sono date da 1-matrici sparse di numeri con segno;

• i complessi p (come insiemi di celle, o basi di spazi lineari, o trasformazioni lineari graduate) sono rappresentati da 2 matrici sparse di numeri con segno. Hanno una complessità spaziale inferiore rispetto alle strutture dati comuni di efficienza ben nota. Quindi lo stoccaggio richiesto da Space(A) è pari a 2/3 del mezzo spigolo, largamente utilizzato in geometria computazionale, e 1/2 del bordo alato , spesso usato come rappresentazione di riferimento per la modellazione solida delle varietà [1]. La cardinalità di tutte le relazioni di incidenza/adiacenza tra cellule p e cellule q (1 ≤ p,q ≤ 3) anche è minima. Perciò, ogni serie di domande locali sulle incidenze/adiacenze 3×3 tra cellule p può essere risolta con moltiplicazione, tramite kernel software per prodotto a matrice sparsa e trasposizione, semplicemente raccogliendo i vettori di coordinate di catene di unità, “soggetto” di interrogazioni elementari, come colonne di una matrice sparsa Q, e moltiplicando a sinistra Q per uno/due operatori matrici [∂1] e/o [∂2], opportunamente ordinate e/o trasposto, per ottenere l'equivalente algebrico di più query di database contemporaneamente.

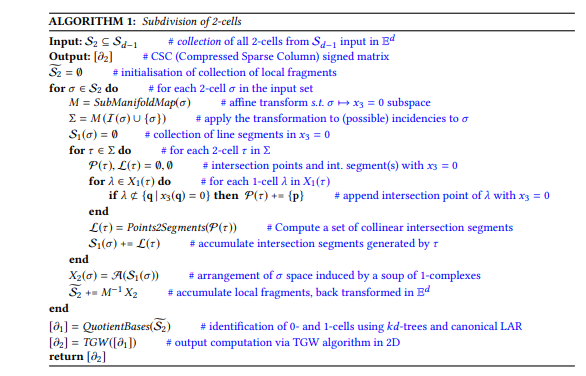
**3 Algoritmi di arrangiamento basati su catene**

In questa sezione, forniamo uno pseudocodice leggermente semplificato dei principali algoritmi introdotti nella sezione precedente e ne discuteremo la complessità nel caso peggiore. Un'implementazione, molto simile a questo pseudocodice, è disponibile come pacchetto Julia open source in Github . Lo stile dello pseudocodice è una miscela di stili Python e Julia. Alcune parole sulle notazioni: le lettere in greco sono usate per le celle di una partizione spaziale e le lettere romane per le catene di celle, tutte codificate come interi con segno o matrici sparse di interi con segno. Al fine di fornire una scrittura formalizzata degli algoritmi pseudo-codificati in questa sezione, dobbiamo introdurre le seguenti convenzioni. [∂d ] o [c] stanno per matrici generali o matrici colonna, rispettivamente, mentre ∂d [h, k] o c[σ] rappresentano i loro elementi indicizzati. Inoltre, |c| sta per indici senza segno (diverso da zero) della matrice sparsa [c]. L'istruzione di assegnazione accumulata A += B sta per A = A + B, dove il significato del simbolo “+” dipende dal contesto, ad es. può stare sia per somma (di catene), sia per unione (di set) o per concatenazione (di colonne di matrice). Analogamente, A -= B sta per A = A - B.

**3.1 Disposizioni di 2 celle (algoritmo Merge)** La sequenza di calcoli eseguiti su ciascuna 2 celle σ ∈ S2 ⊆ S è discussa di seguito.

**3.1.1 Frammentazione di una 2-cella.**

In una prima fase, il sottoinsieme I(σ) di 2 celle potenzialmente intersecanti viene calcolato dall'intersezione dei risultati di tre query sul riquadro σ di delimitazione, rispetto ai tre alberi di intervallo 1D generati all'inizio della pipeline. L'albero dell'intervallo 1D è stato costruito utilizzando uno degli intervalli laterali delle scatole di contenimento 3D delle celle di input 2. In una seconda fase, l'insieme Σ = {σ } ∪I(σ) viene trasformato in modo che σ sia mappato nel sottospazio z = 0 . Le 2 celle mappate vengono utilizzate per calcolare un insieme di segmenti di linea in E2 , generato da intersezione dei bordi di 2 celle con il piano 2D. Coppie alternative di tali punti di intersezione vengono infine unite, lungo la linea di intersezione con z = 0 di ciascuna 2-cella. L'elaborazione planare della 2 celle continua intersecando a coppie tutta la linea calcolata di segmenti e producendo un grafico lineare. I bordi vengono rimossi e viene calcolato il massimo dei sottografi connessi a 2 vertici, con l’algoritmo di Hopcroft e Tarjan. Solo i componenti bi-connessi non esterni entrano nei seguenti calcoli, poiché le altre parti del grafo sono esterne o certamente penzolanti (sottografi 1-collegati), e contribuiranno separatamente alla disposizione degli spazi. Infine, la 2-catena orientata della partizione A(Σ) è calcolata utilizzando il TGW in 2D, generando così la matrice ∂2(σ) da X(σ). Il processo di frammentazione viene ripetuto per ogni σ ∈ S2, con ogni mappa geometrica µ(σ) : X0(σ) → E 2 composto con la sua trasformazione inversa in Mi 3 .



**3.1.2 Complessità della suddivisione in 2 celle.**

La complessità temporale dell'algoritmo 1 è limitata dal numero n di 2-celle nella raccolta di input Sd−1 volte il costo nel caso peggiore richiesto dal suddivisione di uno di essi. A sua volta, questo costo dipende dalle dimensioni e dalla distribuzione dell'effettivo input, cioè sul numero di 2 celle potenzialmente intersecanti in I(σ). Il calcolo di ogni set I(σ) viene eseguito nel tempo di interrogazione degli alberi di intervallo, ovvero nel tempo O(log n + k), dove k << n è la media lunghezza del risultato. Poiché log n è solitamente dominato da k, possiamo aggiungere questo fattore al nostro limite per la ricerca di tutti gli insiemi potenzialmente incidenti. Pertanto, il calcolo di {I(σ) | σ ∈ S2} per tutte le 2-celle di input è O(kn), con k che dipende dalla densità dei dati, e O(n 2 ) nel caso peggiore k = n. In tutti gli altri casi il numero di 2 celle incidenti in una data cella è limitato da un numero costante k1. Se il numero massimo di 1-celle sul confine di-2 celle è k2, allora l'intero calcolo dell'algoritmo 1 richiede il tempo O(k1k2n + A), dove A è il tempo necessario per calcolare gli insiemi di quozienti, cioè per incollare tutti gli X2(σ) in E D spazio. Quando d = 3, le trasformazioni Qσ di ogni insieme (vedi 2.1) sono calcolabili in tempo O(1); costruire un kd-tree statico generato da m punti richiede O(m log 2m); e ogni query per trovare il vicino più vicino in un kd-tree bilanciato richiede in media un tempo O(log m). Il numero di occorrenze dello stesso vertice su 2-celle incidenti è certamente delimitato da una piccola costante k3, approssimativamente uguale a m/v, dove v = #X0 è il numero di celle 0 dopo l'elaborazione dell'identificazione. La trasformazione dell'output in forma canonica (array 1 ordinato di interi) viene eseguito in O (1) per ciascun arco, quindi dando A = O(m log2m) + O(m log m) + O(1) = O(m log m). In conclusione, il tempo di esecuzione nel caso peggiore dell'algoritmo 1 è O(kn + k1k2n + m log m) = O(n(k + k1k2) + m log m), degenerando in O(n 2 ), quale è noto per essere il limite del caso peggiore per la rimozione della linea nascosta.

**3.2 Calcolo degli insiemi di quozienti (algoritmo di congruenza)**

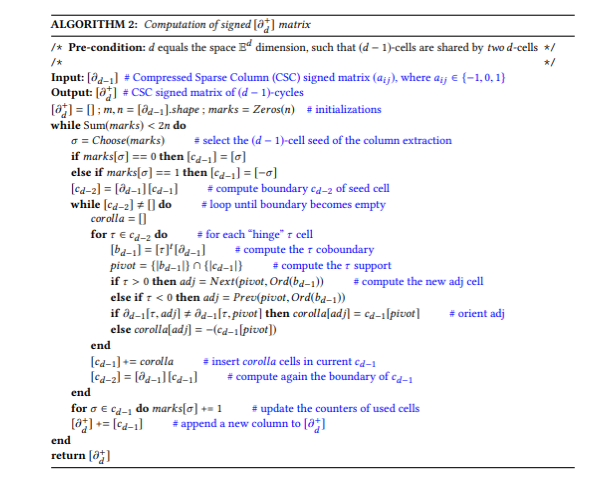
Piccole matrici sparse di operatori con segno ∂2(σ) : C2(σ) → C1(σ) sono già state assemblate indipendentemente in 2D per ogni frammentato, cioè per ogni X 2 , come dettagliato nella precedente Sezione 3.1.1. L'output di quella fase della pipeline è una raccolta ƒ := {C•(σ), σ ∈ S2} di piccoli complessi a catena, uno per ogni ingresso a 2-celle, incastonato in Mi 3 . Sono stati costruiti applicando ripetutamente in 2D il TGW algoritmo (vedi Sezione 3.3) e mappando i risultati in 3D. Si calcolano a questo punto le relazioni di p-congruenza, partendo da p = 0. Pertanto, le catene di unità 0 sono identificate numericamente tramite le loro mappe geometriche e arrotondate a scatto con numeri identificazione di punti vicini coincidenti usando un kd-tree. L'unità congruente 1-catena e 2-catena sono identificati simbolicamente, facendo uso della loro unica rappresentazione canonica indicizzata. La rappresentazione canonica di una catena d unitaria è l'array di indici ordinati degli elementi unitari del suo (d − 1)-ciclo. Le 2-celle dell'uscita del complesso X2(S2)sono incastonate in E 3 , scritto come 1 cicli, cioè, come combinazioni lineari di 1-celle con segno, memorizzate per colonna nella matrice dell'operatore ∂2 : C2 → C1. Una cella τ , è scritta per convenzione come 1u ik 0 − 1u io 0 quando k > h. Le regole convenzionali utilizzate in questo documento sul segno e l'orientamento delle celle sono: riassunti alla fine della Sezione A.1.2.

**3.3 Calcolo di ∂2 e ∂3 (algoritmo TGW)**

* **3.3.1 Confezione regalo topologica**. L'algoritmo è stato introdotto nella Sezione 2.3. Qui forniamo uno pseudocodice leggibile, con l'unica avvertenza che effettivamente calcola un insieme ridondante di generatori per C3 (risp. C2), come 2 cicli collegati minimi (risp. 1-cicli) da una matrice ∂2 (risp. ∂1). Il dato pseudocodice fa uso di simboli matematici e di operazioni matematiche di alto livello; l'effettiva implementazione in Julia utilizza array sparsi e coordinate discrete in {−1, 0, 1}, per ottenere un'esecuzione efficiente in termini di spazio di archiviazione e tempo di calcolo. Notare la pre-condizione dell'algoritmo 2, avvertendo che il metodo utilizzato calcolerà la d matrice solo per una decomposizione cellulare del d-spazio. Infatti, solo in questo caso le (d − 1)-celle sono condivise da esattamente due d-cellule, inclusa la cella esterna. Questa condizione implica che il complesso cellulare di input si applica a quello che dovrebbe essere un complesso CW (possibilmente non connesso), con tutte le celle omeomorfe a sfere. Si noti anche che la matrice dell'operatore di confine per lo spazio della catena d di un cellulare complesso con fori e componenti interni ed esterni sarà costruita a partire dall'uscita dell’ algoritmo 2 in una fase successiva. Il predicato di terminazione dell'algoritmo 2 è una conseguenza della proprietà sopra: l'algoritmo termina quando tutti i numeri di incidenza nella matrice dei segni sono 2, quindi che la loro somma è esattamente 2n, dove n è il numero di (d − 1)-celle, pari al numero di colonne nella matrice di input [∂d−1].
* **3.3.2 Input valido e output univoco**. L'algoritmo funziona correttamente con un input legittimo. In particolare, gli scheletri in ingresso (d − 1) devono essere regolari, cioè senza parti penzolanti, in modo che ogni (d − 1)- la cellula appartiene al massimo a due (d − 1)-cicli. In 2D, questo fatto è garantito applicando l'algoritmo separatamente a ciascun componente 2-massimo connesso dell'1-scheletro, considerato come un grafico, e poi unendo i risultati (chiaramente disconnessi). Analogamente, in 3D, il grafico delle adiacenze di 2-celle non devono contenere sottografi penzolanti. L'insieme di validità dell'input può contenere 2-scheletri di 3-complessi, confini di modelli solidi, insiemi di componenti al contorno molteplici di modelli solidi non molteplici. Naturalmente, applicare l’algoritmo ai dati che non determinano una partizione dello spazio di incorporamento non ha senso e produce un risultato vuoto. Quando applicato a un input valido, come descritto sopra, l'algoritmo TGW è sempre corretto, perché produce sempre l'insieme di generatori per Cd che soddisfa l'equazione di seguito: [∂d ] = (aij) dove # Õ Xd-1 i=1 # Xd j=1 | aij | = 2 (#Xd−1).

I risultati sono anche unici, poiché altrimenti due diverse basi produrrebbero due operatori di confine che, applicati alla 3-catena totale (vettore di tutti quelli) restituirebbe lo stesso ciclo di confine, il che è impossibile. Non ci sono ambiguità nell’ algoritmo, poiché in un d-complesso ogni due d-celle condividono al massimo due (d − 1)-celle, o esattamente due se viene considerata la cellula d esterna. Inoltre, si ferma quando viene esattamente raggiunta quest'ultima condizione. Notare che la scelta opportuna dei successivi “petali” da “corolla” (vedi lo pseudocodice in Algoritmo 2) implica che una 2-cella non può essere utilizzata più di due volte.

* **3.3.3 Complessità dell'estrazione a 3 celle**. In tre dimensioni, l'algoritmo 2 costruisce in modo iterativo (outer while) un'unità 3-catena (rappresentata come un 2-ciclo, cioè come una 2-catena chiusa), costruendo la colonna corrispondente della matrice [∂ + 3 ], e quindi aggiungendo una colonna di confine esterna per ciascuna componente connessa del complesso di input, come dettagliato in 3.4. La complessità spaziale di una 3-cella è misurata da un insieme di triple (riga, colonna, valore) implementate come tripla di array (I, J, Values) per valori diversi da zero in una colonna a 3 celle, ovvero con la sua rappresentazione come ciclo di unità 2-catene. Quindi, il numero totale di triple, cioè la complessità spaziale del COO rappresentazione di [∂ + 3 ], è esattamente 2n, dove n è il numero di 2 celle nello scheletro X2. La costruzione di un singolo 3 celle richiede la ricerca dell'adiacente 2 celle adj per ogni unità pivot 2-catena nel guscio di confine. La ricerca di next o prev 2-cell come adj per ogni pivot richiede l’ ordinamento circolare di questo sottogruppo di permutazione di 2 celle incidente a ciascuna 1 cella su ciascun confine di un 2-ciclo incompleto. Un semplice ordinamento circolare di 2 celle, utilizzando gli angoli tra le normali ai loro piani, funzionerebbe sempre solo con 2 celle convesse; nel nostro caso, poiché 2 celle possono essere non convesse, questo ordinamento può andare storto quando si calcola la normale alla faccia con il terzo vertice mal scelto. Per questo motivo è necessaria una CDT (Triangolazione Constrained Delaunay) di ogni incidente a 2-facce, che è implementata utilizzando la libreria Triangle, portata nel linguaggio Julia. Di conseguenza, abbiamo diversi tipi di insiemi triangolari attorno a un bordo, dove ogni insieme è delimitato da un piccolissimo intero, quindi ogni ordinamento è O(1) nel tempo. Il numero totale di tali tipi è delimitato in alto da numero di (d − 1)-celle sul confine d-cell (uguale a 6 per i 3-complessi cubici e a 4 per simplicial 3-complessi, e ad un piccolo intero in generale). I sottoinsiemi da ordinare sono codificati nelle colonne della matrice di incidenza da 2 celle a 1-celle, cioè dagli indici i, j di elementi non nulli di [∂2]. Il calcolo (senza segno) di [∂2] può essere eseguito mediante moltiplicazione SpGEMM14 di due matrici sparse, quindi in tempo lineare con la dimensione dell'output, cioè con il numero di elementi diversi da zero della matrice [∂2]. Riassumendo, se n è il numero di d-celle e m è il numero di (d − 1)-celle, la complessità temporale di questo algoritmo è O(nm log m) nel caso peggiore di complessità illimitata di d-cellule, e grosso modo O(nk log k) se la loro complessità (d − 1)-ciclo è limitata da k.



**3.4 Shell isolati (algoritmo dei fori)**

In generale, sia la cella esterna che le celle interne possono contenere fori isolati e/o isolati componenti, cioè sottocomplessi i cui confini esterni non si intersecano tra loro. Parlare di fori isolati non è del tutto esatto, poiché i fori non sono mai vuoti all'interno di una disposizione, cioè una partizione dello spazio ambiente, ma contengono un componente isolato all'interno del loro confine rappresentato come a (d − 1)-ciclo. Lo scopo di questa sezione è discutere la gestione dei componenti isolati e la loro confini, da considerare buchi all'interno del loro spazio contenitore. L'algoritmo TGW produce complessi CW, nonostante il fatto che la suddivisione di Merge può generare cellule d non contrattili, cioè cellule con buchi. Questi spazi sono gestiti combinando complessi CW standard, cioè con cellule omeomorfe a sfere, e aggiungendo d-cellule all'interno delle cellule D. In altre parole, il confine dei fori in una cella proveniente da sub-complessi disconnessi viene unito nella cella contenitore. L'orientamento viene gestito in base alla parità del relativo rapporto di contenimento. La gestione dei confini isolati riguarda essenzialmente il aggiunta/rimozione di colonne nella matrice di confine finale.

**3.4.1 Sintesi dell'algoritmo Holes:**

Dobbiamo considerare due questioni principali:

(a) il calcolo delle componenti massime connesse di Xd−1 può produrre h > 1 componenti d scollegate di output complesso Xd ;

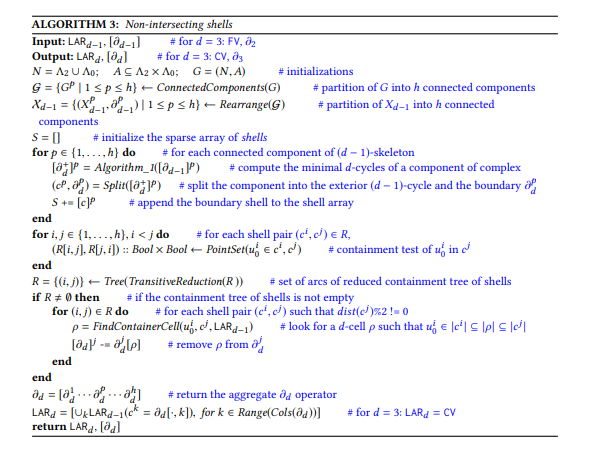
(b) l'inclusione di componenti all'interno di singole celle del complesso d di output. Di seguito elenchiamo le fasi principali dell'algoritmo Holes per occuparsi di questi problemi. Il nostro obiettivo è il calcolo sia dello scheletro Xd, sia dell'operatore d per spazi con multipli componenti annidati nei fori. Nota che si applicano esattamente gli stessi punti (ridimensionati nelle dimensioni) prima e dopo l'esecuzione di TGW, per entrambe le disposizioni locali in 2D generate da ogni ingresso a 2 celle σ, e la disposizione globale in 3D generata dall'intero insieme di complessi di input, rispettivamente. Questo dà infatti spunto per una possibile estensione multidimensionale.

**3.4.2 Shell non intersecanti**

Se l'insieme di shell S , , allora le componenti al contorno isolate h (0 ≤ p ≤ h) in S devono essere confrontati tra loro, per determinarne l'eventuale contenimento relativo, e di conseguenza il loro orientamento. La matrice binaria e antisimmetrica h × h M = (mij) del la relazione si costruisce, calcolando ogni elemento mij (i < j), con un singolo punto-ciclo di emissione del raggio, perché cj come esterno/interno, e quindi il suo orientamento relativo è dato dalla parità di c J in R. Quando la cancellazione delle celle vuote è stata eseguita per tutti gli archi “solidi” di R, la matrice aggiornata [∂d ] P può essere assemblata nella matrice di operatori ∂d finale.

3**.4.3 Complessità della gestione della shell**

Il calcolo delle componenti connesse di un grafo G può essere eseguito in tempo lineare . Il riconoscimento delle shell h richiede il calcolo di [∂ + D ] P (1 ≤ p ≤ h) e l'estrazione del confine di ogni componente connesso X P D . Per calcoliamo la relazione ridotta R eseguiamo O(h 2 ) prove di contenimento punto-ciclo, lineari nelle dimensioni di un ciclo, quindi impiegando un tempo O(h 2n), con h numero di conchiglie e n dimensione media a ciclo. In realtà, il test di contenimento punto-ciclo può essere facilmente calcolato in parallelo, con una trasmissione minima sovraccarico degli argomenti. La ristrutturazione delle sottomatrici di confine ha lo stesso costo della lettura/riscrittura di colonne di una matrice sparsa, a seconda del numero di non zero di [∂3], e quindi è O(n#Xd ), cioè lineare con il prodotto del numero di d-celle e la loro dimensione media n come catene di (d − 1)-cellule, con n dimensione del ciclo isolato medio.



**3.5 Il quadro completo**

Una breve sintesi dei passaggi sequenziali dell'intera pipeline computazionale, dalla raccolta degli input a uscita complessa a catena, segue nell'impostazione più generale, con entrambe le componenti isolate (entro la cella esterna) e componenti isolati possibilmente annidati (all'interno di fori nelle celle interne).

**Input** Facet selection, cioè costruzione della collezione Sd−1 da Sd , usando LAR. **Indicizzazione** Indice spaziale formato dall'intersezione di d alberi di intervallo su riquadri di delimitazione di σ ∈ Sd−1. **Decomposizione** Intersezione a coppie di segmenti di linea in σ ∪ I(σ), per ogni σ ∈ Sd−1. **Congruenza** Basi graduate delle classi di equivalenza Ck (Uk ), con Uk = Xk /Rk per 0 ≤ k ≤ 2. **Collegamento** Estrazione di (X P d-1 , p d-1 ), componenti massimali connessi di Xd−1 (0 ≤ p ≤ h). **Basi** Calcolo della base del ciclo ridondante [∂ + D ] P per ogni componente p, tramite TGW. **Confini** Accumulo in H += [o] P (hole-set) del ciclo di confine esterno da ogni [∂ + D ] P . **Contenimento** Calcolo della relazione di contenimento antisimmetrico S tra [o] p fori in H. **Riduzione** Riduzione R di S e generazione di foresta di alberi piatti ⟨[od ] P , [∂d] P . **Adiacente** di radici [od ] R a (unica) cella esterna e non-radici [∂ + D ] Q alle celle contenitore. **Assemblaggio** Assemblaggio quasi diagonale a blocchi di matrici relative a componenti isolati [∂d ] P . **Output** Mappa di confine globale [∂d ] di A(Sd−1) e ricostruzione di catene 0 di d-celle in Xd .

**4 Confronto con altri approcci**

In questa sezione menzioniamo alcune connessioni rilevanti del presente approccio con lavori recenti su argomenti simili e discuteremo alcune osservazioni in relazione al nostro lavoro. In precedenza abbiamo introdotto un approccio topologico all'omologia per sottoclassi di spazi suddivisi costituito da mappe combinatorie e generalizzate. La mappa generalizzata (Gmap) è un modello combinatorio che permette di rappresentare e maneggiare oggetti suddivisi tramite “connecting darts” tra coppie di cellule. Le Gmap sono usate per descrivere la topologia di oggetti cellulari di tipo molteplice dove le cellule p sono omeomorfe alle sfere p.

Alayrangues, Damiand, Lienhardt e Peltier danno un algoritmo per costruire mappe di confine per 0 ≤ i ≤ 3, concentrandosi sull'equivalenza tra calcolo dell'omologia tramite Gmap e complessi simpliciali. La complessità temporale delle mappe di confine è lineare nel numero di numeri di incidenza. Noi abbiamo già ottenuto lo stesso risultato, lineare nella dimensione dell'output sparso, per il calcolo tramite SpGEMM moltiplicazione, quando il complesso di input è noto.

La complessità di TGW per il calcolo dell'incognita Xd = A(Sd−1) è ovviamente maggiore e eguaglia lo standard nella modellazione solida.

La differenza principale con il nostro approccio è che Alayrangues e colleghi partono da un dato modello cellulare Gmap, la cui costruzione è piuttosto complessa e richiede operazioni interattive con un'interfaccia utente grafica o un sistema di logica simbolica (come come INRIA's Coq) con un linguaggio di specifica formale. Se la semplicità metrica conta, la rappresentazione algebrica di catene con array sparsi si confronta bene con le catene di Gmap. Con Selective Geometric Complex , Rossignac e O'Connor hanno proposto un'estensione significativa dei complessi CW topologici, costruendo p-cellule con strutture di dimensione 0 ≤ k < p all’interno delle cellule, cioè non necessariamente incorporate in un confine cellulare. Il bit di selezione associato a ciascuno cella consente la scelta selettiva delle sottostrutture. L'associazione di incidenza e ordinamento locale tra le celle incidenti sono mantenute tramite collegamenti gerarchici, analoghi al diagramma di Hasse tra i vertici, bordi e facce, nonché con estensioni geometriche di patch di superficie non lineari. Due attributi per c.boundary e c.star di una cella restituiscono le celle sul suo confine e quelle di cui è al confine. La ricerca del confine di sottostrutture più complesse è algoritmica. Nel presente documento, usiamo invece operatori lineari graduati e combinabili per il confine e co-confine per attraversare, sia a livello locale che globale, la gerarchia di incidenza. Quindi otteniamo, tramite moltiplicazione di matrici sparse e vettori, una caratterizzazione lineare completa dello spazio topologico. Anche gli aggiornamenti locali alla topologia, tramite operatori di Eulero, possono essere fatti algebricamente . Un’ altra differenza significativa riguarda la grande quantità di informazioni e indicazioni associate da SGC a ogni cella, inclusi estensione, dimensione, confine, bit di attività e attributi estensibili. Al contrario, nel presente lavoro una catena p unitaria orientata è caratterizzata solo da un indice intero con segno a Up base, e dal suo ciclo di confine con segno (p − 1), memorizzato come una colonna sparsa in [∂p ]. Ovviamente tutte le interrogazioni topologiche, sia locali che globali, sono consentite da opportune moltiplicazioni SpGEMM. A noi consentono di annidare cicli interni (fori) e sottostrutture nelle celle, ma non per il contenimento esplicito di bordi e punti interni.

La maggior parte degli algoritmi topologici è di natura algebrica,quello di Zhou, Grinspun, Zorin e Jacobson, calcola le disposizioni delle mesh per la geometria solida, prende come input un numero qualsiasi di mesh triangolari, risolve le intersezioni triangolari in 3D e assegna un vettore numerico di avvolgimento a celle suddivise, per valutare espressioni booleane variadiche. I loro i dati sono rappresentati da (piccoli) BSP arricchiti con patch superficiali convesse esplicite sui nodi, e struttura di adiacenza tra i nodi, insieme a una grande quantità di informazioni aggiuntive. L'approccio può essere suddiviso in due fasi: in primo luogo, l'aggiunta iterativa di mesh a un arrangiamento; secondo, eseguendo tutte le classiche operazioni booleane. Al contrario, non aggiungiamo ogni input al risultato precedente ma, in fase di scomposizione, operano indipendentemente su ogni ingresso a 2 celle, secondo un approccio basato sui dati paralleli. È anche notevole che il presente approccio funziona con mesh più generali: insiemi di 2-varietà con e/o senza confine, insiemi di non-varietà, insiemi di 3-varietà, ecc., contro solo insiemi di mesh triangolari. Noi non ne discuteremo qui, ma estendere il nostro approccio alle operazioni booleane e alle funzioni booleane è semplice. Un algoritmo di riparazione della mesh estremamente veloce con topologia garantita è principalmente basato sull'aritmetica in virgola mobile e richiede un'aritmetica esatta solo in relativamente poche situazioni. Diversamente da Attene , non distinguiamo tra caso molteplice e non molteplice, e non utilizziamo una struttura dati speciale in nessuna fase della pipeline, ad eccezione di alberi di intervallo 1D e alberi di kd per l'accelerazione. Identificando le condizioni che rendono l'aritmetica in virgola mobile non affidabile.

J.R. Shewchuk, l'autore della libreria Triangle, utilizzata nel nostro metodo, identifica la chiave per predicati geometrici veloci e robusti nell'aritmetica a virgola mobile di precisione adattiva. In effetti, il i risultati numerici che abbiamo ottenuto su triangolazioni con grandi disposizioni di segmenti di linea 2D sono molto veloci.

Abbiamo portato le funzioni CDT (Constrained Delaunay Triangulation) dalla sua libreria C al linguaggio Julia, e l'abbiamo usato per triangolare al volo ogni 2-cella non convessa, al fine di calcolare l'ordinamento delle 2 celle "corolle" attorno alle 1 celle "pivot" nel TGW 3D.

Campen e Kobbelt presentano una tecnica per implementare operatori che modificano la topologia di mesh poligonali alle intersezioni e alle auto-intersezioni, combinando un octree adattativo con partizioni spaziali binarie annidate. Un'analoga tecnica decompositiva è stata introdotta nel linguaggio geometrico PLASM di Scorzelli, Paoluzzi e Pascucci: la tecnica viene ora sostituita dai metodi indicati nel presente documento, poiché non garantisce sufficiente robustezza e velocità.

Guibas e Marimont descrivono un algoritmo dinamico per calcolare la disposizione di un insieme di segmenti di linea nel piano digitale e per agganciare i punti di intersezione al centro dei pixel. Scattiamo piccoli ammassi di punti molto vicini (numericamente “quasi congruenti”), con un arrotondamento al centro dei loro ϵ-quartieri in 2D, con diametro medio di 10−16, vicino alla risoluzione della virgola mobile binaria.

Barki, Guennebaud e Foufou presentano un metodo esatto, robusto ed efficiente per eseguire operazioni booleane regolarizzate su mesh 3D generali. Usano una triangolazione di tutte le facce e riducono l'intersezione di due superfici all'intersezione 3D di due triangoli. La loro semplice decomposizione, il processo per i volti che si intersecano, è molto simile al vecchio articolo di Paoluzzi e dei suoi studenti. Entrambi contengono una procedura per il calcolo delle operazioni booleane regolarizzate comprese le conchiglie isolate. Il punto nuovo su questo argomento è che nel presente documento i gusci orientati verso l'interno, così come i componenti isolati, sono gestiti attraverso la somma firmata dei chiusi di catene e implementate come somma o differenza delle loro coordinate vettoriali in Z o Z/3Z.

Ricordiamo infine che Half-edge, la più piccola rappresentazione efficiente conosciuta della topologia di grafi planari e 2-varietà chiuse di Muller e Preparata, largamente utilizzati in Computational è la geometria per le triangolazioni e i diagrammi di Voronoi.

Con gli operatori [∂1] e [δ1] dati nel presente lavoro, otteniamo per questa classe la dimensione ottimale Ω(n) = 4#E, uguale alla dimensione di input: due vertici e due facce per bordo. È bene noto che entrambe le relazioni EV ed EF pesano per 2#E, cioè uguale allo spazio occupato dalle mappe [∂1] o [∂2] , che consentono anche l'equivalente algebrico di più query di database contemporaneamente.

**5 Passato, sviluppo e prospettive di questo progetto**

I primi tre autori hanno iniziato questo progetto sull'informatica con catene, cicli, co-catene e co-confine in una serie di seminari sui nuovi metodi algebrici per la simulazione fisica e l'ottimizzazione del disegno geometrico. Questo progetto ha ricevuto il premio IBM SUR nel 2003. Array sparse LAR, grandi dati geometrici e servizi geometrici sono stati discussi in molti incontri a Roma, Parigi, Madison, Berkeley e Berlino. Si è data vita a una sequenza duratura di discussioni sul web e faccia a faccia, esperimenti algebrici e software test, producendo negli ultimi anni tre implementazioni parziali open source in Python e Julia. Sono state utilizzate implementazioni parziali per esperimenti basati su software di tracciamento degli utenti e interni geo-mapping in Building Information Modeling (BIM) basato su LAR), meta-progettazione di un ospedale generale, e fornitura di servizi web volti alla decostruzione e al riutilizzo degli edifici.

Attualmente, alcuni degli autori hanno materializzato un pacchetto Julia per disegno topologico e geometrico, inclusa una prima implementazione degli algoritmi in questo articolo.

Una seconda versione include la vettorizzazione sulla GPU e la concorrenza è basata su attività che utilizzano Julia nativa . L’obiettivo futuro è di portare la pipeline delle mappe della catena sul GPX-1 di Nvidia, per fondersi con il deep learning dall'imaging. Si sta già utilizzando l'approccio di modellazione introdotto qui basandosi sul Julian libreria open-source, per lo sviluppo rapido di modelli edilizi dall'analisi del Catasto Italiano documenti e modelli di edifici da immagini 3D scansionate dal volo di droni. Speriamo che anche le strutture di base e gli algoritmi discussi in questo articolo possano trovare un uso appropriato quando combinato con rappresentazioni per reti neurali convoluzionali, basato così come sui tensori e sull'algebra lineare, al fine di combinare correttamente la comprensione dell'immagine e modellazione geometrica. In particolare, per calcolare il complesso della catena di un'incognita, la disposizione dello spazio dovrebbe combaciare bene con le NN profonde. Anche i primi esperimenti con i metodi topologici nell'imaging medico sembrano promettenti.

**6 Sintesi dei risultati e conclusione**

Abbiamo introdotto una nuova visione sul calcolo topologico delle disposizioni spaziali, che potrebbe trovare buon uso in disparati sottodomini di calcolo geometrico e visuale, discusso un originale architettura computazionale basata sull'algebra topologica lineare e possiamo dire che il nostro approccio è in sintonia con le tendenze attuali verso l'hardware ibrido e le sue applicazioni software più avanzate. In particolare, in questo documento forniamo un'implementazione in pseudocodice dell'intera pipeline computazionale: da un insieme di oggetti geometrici virtuali al complesso di catene (Cp, ∂p ) della loro partizione di spazio, dando una caratterizzazione completa della topologia indotta dall'input. Questo risultato si ottiene andando oltre i complessi semplici e lavorando con la topologia generale lineare a tratti con cellule non contrattili. Tra i punti di forza citiamo: la rappresentazione compatta; la combinabile natura delle mappe, consentendo più query sulle relazioni della topologia locale 3 × 3, tramite nuclei sparsi per moltiplicazione e trasposizione; la frammentazione indipendente delle celle di input attraverso la congruenza cellulare; e l'algoritmo topologico dell’algoritmo gift wrapping. Ultimo ma non meno importante, questo approccio può essere estendibile a una dimensione superiore. Crediamo che un'implementazione in tempo reale dei nostri algoritmi su GPU possa generare nuove tecniche per la comprensione delle immagini, in particolare quando gli input provengono da telecamere 3D di nuova generazione, verranno installate su dispositivi mobili a guida autonoma veicoli.

**APPENDICE**

Per comodità dei lettori, ricordiamo qui alcune definizioni e fatti sull'elaborazione con catene e co-catene.. Usiamo la lettera greca per le celle e le lettere romane per le catene, cioè per le combinazioni firmate di celle. Con qualche abuso di linguaggio,spesso sono indicate le cellule in Λp e le catene di unità (singleton) in Cp.

A.1 Complessi di co-catene:

Complesso cellulare. Sia X uno spazio topologico e (X) = Dp (p ∈ {0, 1, . . . ,d}) una partizione di X, con Λp un insieme di p-celle (relativamente) aperte, connesse e molteplici. Una struttura CW sullo spazio X è una filtrazione ∅ = X−1 ⊂ X0 ⊂ X1 ⊂ . . . ⊂ Xd−1 ⊂ X = ∪ p Xp , tale che, per ogni p, lo scheletro Xp è omeomorfo a uno spazio ottenuto da Xp-1 per attaccamento di p-celle in p = Λp (X). Un complesso CW è uno spazio X dotato di una struttura CW, ed è anche chiamato a complesso cellulare. Un complesso cellulare è finito quando contiene un numero finito di celle. Un normale d-complex è un complesso in cui ogni p-cella (p < d) è contenuta nel confine di una d-cella. L due cellule d sono orientate coerentemente quando le loro cellule comuni (d −1) hanno orientamenti opposti. Un cellulare d-complesso X è orientabile quando le sue d-celle possono essere orientate coerentemente. Lo spazio di supporto |σ | di una cella è il suo insieme compatto di punti.

A.1.2 Gruppi a catena.

Le catene sono definite associando coefficienti alle celle. Dal momento che si desidera aggiungere catene, si devono prelevare coefficienti da un insieme dotato della struttura di un gruppo commutativo, o più forte. Sia (G, +, 0) un gruppo commutativo non banale, il cui elemento di identità è indicato con 0. Una p-catena di X con coefficienti in G è una mappatura cp : X → G tale che, per ogni σ ∈ Xp , invertendo l'orientamento di una cella cambia il segno del valore della catena: cp (−σ) = − cp(σ). L'addizione della catena è definita dalla somma dei valori della catena: se cp1 , cp2 sono p-catene, allora (cp1 + cp2 )(σ) = Cp1 (σ) + cp2 (σ), per ogni σ ∈ Xp . Il gruppo risultante è indicato con Cp (X;G). Quando il contesto è chiaro, il gruppo G è spesso lasciato implicito, scrivendo Cp (X). Sia σ una cella orientata in X e g ∈ G. La catena elementare il cui valore è g su σ, −g su −σ e 0 su qualsiasi altra cella in X è indicata con gσ. Ogni catena può essere scritta in modo univoco come somma di catene elementari. Si pensa spesso che le catene colleghino l’orientamento e/o la molteplicità alle celle: se i coefficienti sono presi dal gruppo G = ({−1, 0, 1}, +, 0) ≃ (Z/3Z, +, 0), quindi le celle possono essere solo scartate o selezionate, eventualmente invertendo il loro orientamento. Un ciclo p è una catena p chiusa, cioè una catena p senza confine. È utile per selezionare un orientamento convenzionale per orientare automaticamente le celle. Le 0-celle sono considerate tutte positive. Le p-celle chiuse possono ricevere un orientamento coerente (interno) in accordo con l'orientamento della prima (p − 1)-cella nella loro rappresentazione canonica ordinata sugli indici dei loro (p − 1)-cicli. Quindi una cella d può essere orientata come il segno del suo volume orientato.

A.1.3 Spazi catena. Per consentire non solo l'aggiunta di catene, ma anche la combinazione lineare di catene, i coefficienti dovrebbero essere presi da un insieme dotato della struttura di un campo, come (F, +, ×, 0, 1), dove 0 e 1 indicano rispettivamente le identità additiva e moltiplicativa. Le unità di catene sono catene elementari il cui valore è u = 1σ per qualche cella σ. Ogni catena può essere scritta in a modo unico come combinazione lineare di catene unitarie u ∈ U , se non si tiene conto della cella esterna. Quindi, lo spazio delle p-catene Cp è dotato di una base standard (o naturale), costituita da tutte le catene p di unità indipendenti. In particolare, #Ud = #Λd− 1. Spesso, con qualche abuso di notazione, non si fa distinzione tra una cellula p e la corrispondente catena p dell'unità.

A.1.4 Matrici caratteristiche. Dato un insieme S = {sj }, la funzione caratteristica χA : S → {0, 1} assume valore 1 per tutti gli elementi di A ⊆ S e 0 per tutti gli elementi di S non in A. Chiamiamo matrice caratteristica M di un insieme di sottoinsiemi Ai ⊆ S (i = 1, . . . ,n) la matrice binaria M = (mij), con mij = χAi (sj). Una matrice Mp , le cui righe sono indicizzate da catene di unità p e le colonne sono indicizzate da catene di unità 0, fornisce un'utile rappresentazione di una base per lo spazio lineare Cp . Permutare (reindicizzare) o righe fornisce una base diversa. Inoltre mentre le catene sono per lo più presentate come somme formali di celle, nell'implementazione effettiva i loro vettori di coordinate con segno sono usati come array sparsi, e in particolare come mappe CSC (Compressed Sparse Column): N → {-1, 0, 1}.

A.1.5 Spazi di co-catene. Le co-catene sono duali alle catene: le p- co-catene mappano linearmente le p-catene al campo sottostante F. Unità di p- co-catene, che producono 1 quando valutate su una p-catena unità e 0 quando valutate su tutte le altre, formano la base standard dello spazio delle p- co-catene Cp . Gli spazi lineari Cp e Cp , essendo isomorfi, possono essere identificati tra loro in infiniti modi. Diverse identificazioni legittime, pur influenzando le proprietà metriche del complesso catena- co-catena , non cambiano la topologia dei complessi finiti. Poiché useremo solo le proprietà topologiche di complessi finiti catena- co-catena definiti da complessi cellulari lineari a tratti nello spazio euclideo, è possibile scegliere l'identificazione più semplice possibile, consistente nell'identificare ogni elemento della base standard di Cp con il corrispondente elemento della base standard di Cp . In questo documento, noi diamo per scontato che catene e co-catene vengano identificate in questo modo banale.